EXHIBIT D

REDACTED

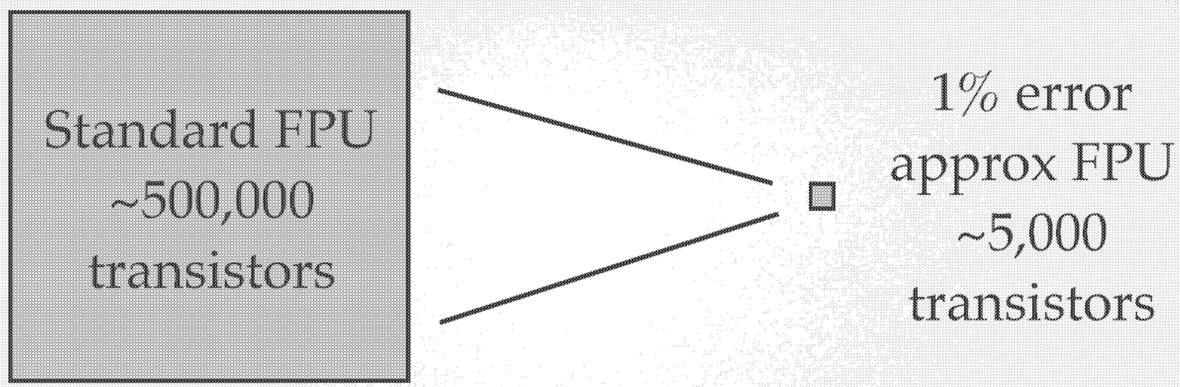
APPLICATIONS / MARKETS / AND DEALS

June 2011

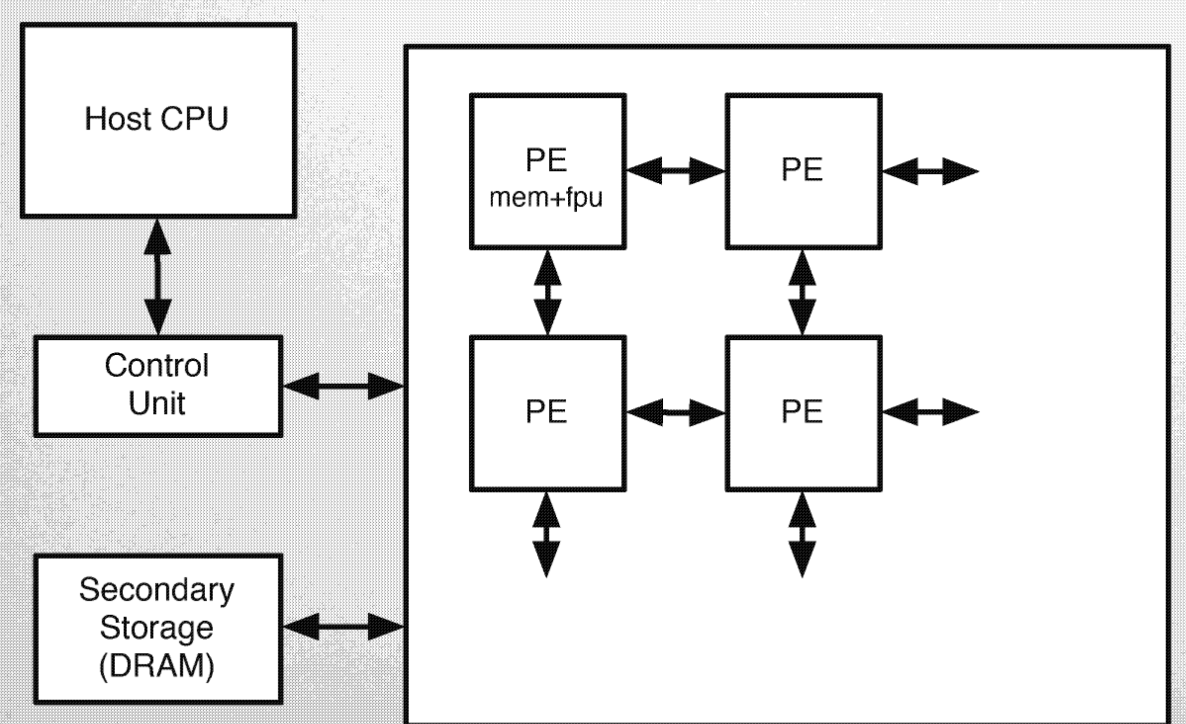
Joe Bates
Singular Computing LLC
MIT Media Lab, Visiting Scientist
Carnegie Mellon CS Department, adjunct prof

joe@singularcomputing.com

REVIEW



Combine FPU with 200 words memory. Build 2D grid, local wires, SIMD



In volume:

4K cores, 100 Gflops, 1 watt, $3

1M cores, 100 Tflops, $5000
(top 100 supercomputer)

REVIEW

LIMITATIONS

- SIMD
- memory per core
- local communication within chip
- i/o bandwidth off chip
- approximate floating point

all but last were extensively studied in 80s:
MPP, MasPar, CM, DAP, ...
so there is much work to draw on,
familiar programming tools (e.g. C parallel C),*
and approx floats often easily managed in software

APPLICATIONS, MARKETS, CUSTOMERS

- Nearest neighbor
  (Sebastian, Shumeet, Hartmut say valuable)
- Optimization (LP, MIP)
- Defense (vision)
- Consumer products (cheap vision)

K-NEAREST NEIGHBOR

BASIC METHOD

- Example applications
  - Image search - want long vectors (1K), big database (10G vectors)
  - Vision - feature correspondence for tracking - fast, low energy

- Brute force - pre-load ~1 database vector per core
  - CPU broadcasts query vector + distance computation instructions
  - find minimum, passing (distance, index) pair
    across rows to right column, then down column, then out

- Time for query broadcast ~ vector length x few ~ 1000 cycles
  Time for extracting minimum ~ grid edge length ~ hundreds
  $\Rightarrow$ ~thousands of times speedup over CPU

K-NEAREST NEIGHBOR

BAD ARITHMETIC

- Have chip find several best of many thousands, then CPU picks nearest from these few
- Results for: chip finding single best / top two / top three

vector len	db size *(one chunk)*	uniform(0,1)	normal(0,1)
100	64K	91.3 / 98.5 / 99.8	92.1 / 99.0 / 99.8
200	32K	88.5 / 97.8 / 99.6	88.8 / 98.0 / 99.7
800	8K	78.7 / 92.9 / 97.7	79.0 / 93.5 / 97.8

Limits of arithmetic can be overcome - little extra time

K-NEAREST NEIGHBOR

SCALING UP

- If big database:
  - load as series of chunks (as much as fits per chunk)
  - search each chunk for best
  - have CPU choose best of those bests
- Chunk loading slow, but if enough queries then OK - query time per chunk can exceed load time per chunk
- e.g., image search for many simultaneous users - yields plenty of queries - thousands per 10 ms batch
- *Limits of memory per core and bandwidth to CPU can be overcome*
- Large database still too slow with 1 chip, but like CPUs, hashing can divide db/queries over many chips

OPTIMIZATION

- Why I said:
  "tech risk reduced enough to focus on business risk"
- Argonne runs Network Enabled Optimization Server
  ~20 varieties of optimization
- Director says by far greatest commercial importance: LP, MIP
- MIP can be reduced (branch & bound)
  to solving sequence of LP problems
- So key question: can we solve large, sparse LP?
  (eg 1M vars, 1M constraints, .1% non-zero)

- Remember Merrick Furst (now Georgia Tech)?
- "Reflex" - new interior point method
- Speed/quality benchmarks - as good as simplex, other ip methods
- Method:
  - have interior point:
    pick random direction, bounce off walls bunch of times
  - alternate: bounce toward goal (until little progress)
    get unstuck (bounce away from prior point)
- Key step is bounce - where does ray intersect nearest plane?
- Method: inner product ray with every plane, to find
  distance to all intersections, choose nearest one
- Like kNN - many local ops (inner prods), then min across chip

- Key requirement - many sparse inner products
- Merrick says can be approximate - random motion is intended
- Not yet definitive,
  but believe I know how to pack data into chip's memory,
  efficiently do sparse approximate inner products,
  and update bouncing point based on results
- Rack sized machine (100 chips) should do large, sparse LP
- So LP promising, so MIP promising,
  so two biggest commercial needs in optimization
  are likely technically solvable

OPTIMIZATION MARKET

- Stand-alone optimization - IBM bought ILOG/CPLEX ~$340M
- Integrating optimization into ERP apparently killed stand-alone
  (from Harvard business school manufacturing professor,
  and history of i2, a previous stand-alone leader, Wikipedia)
  - SAP, Oracle now are leaders
- Brief look suggests market may be 10x
  ~$B scale
- Doesn't include CPU use ("5-10% total CPU at large oil company"),
  home-grown software, or market growth due to cheaper/faster

VISION

FEATURE BASED TRACKING

- Extract significant features
  (each is a feature vector, located at specific point)

- Feature correspondence across frames -
  similar method to nearest neighbor search
  (keep all vectors from last frame on chip,
  search them for similarity to this frame's features)

FEATURE EXTRACTION

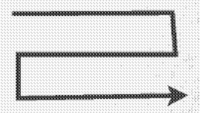
BASIC METHOD

- Load image: time ~ image size
  eg 300x300 image ~ 100K pixels ~ 100K cycles
- Compute enough features to take at least similar time
  eg 100 features, 1000 (parallel) cycles each (~100K cycles)
- Read out most significant feature vectors
  eg 500 vectors, each 200 words long (~100K cycles)

FEATURE EXTRACTION

KERNEL AND HISTOGRAM FEATURES

- Kernel features
  - Broadcast instructions to shift images left-right up-down (snake-like) using local connections
  - As pixels pass cores, broadcast kernel weights, accumulate weighted sums in cores
  - Total time for kernel at all pixels ~ kernel size (eg 11x11)



- Histogram features (eg SIFT, HOG - histogram of oriented gradients)
  - Compute local kernels, eg edge detectors
  - Shift results across region of interest (snake pattern)
  - Accumulate histograms in each core (cores do indirect addressing)
  - Time ~ region size
- Multiscale
  - Form Gaussian blurs similar to kernel computations

FEATURE EXTRACTION

READOUT

- Can't afford to read out feature vector (200+ bytes) from every pixel
- But only ~500 are significant
- Sort them by significance toward some edge
  (known method based on parallel mesh bubble sort)
  Time ~ (# cores along edge) * (feature vector length) ~ 20K cycles
- Read out top 500 (which now are near edge)
  Time ~ 500 * (feature vector length) ~ 100K cycles
- Estimated <u>total</u> time for image input, feature extraction, and readout
  <1M cycles @ 30MHz ⇒ >30 fps (~100Gflops), 1 watt (mobile chip $3)

CUSTOMER DISCUSSIONS

1. BIG DEFENSE CONTRACTOR

- Big flying camera: 400 cell phone cameras $\Rightarrow$ gigapixel frames
- Vision pipeline:
  - per pixel processing (eg color, contrast adjustment)
  - local region processing (kernels)
  - feature extraction (kernels and local histograms)
  - feature correspondence (~nearest neighbor)
  - then higher levels (feature-based tracking) on CPUs

  (this all seems likely to work - evidence includes similar Navy project Singular has with small defense contractor)
- Also want:
  - motion detection based on MoG (works - see MIT work)
  - compression (have evidence for JPEG, which they use)

CUSTOMER DISCUSSIONS

1. BIG DEFENSE CONTRACTOR

- Requires 10-100 Tflops
- Now use many hundreds of FPGAs, ASICs, GPUs, and CPUs
- Size important, power very important (not cost)
- Current evaluation suggests ~10 large, slow Singular chips
- Yields 10x overall power improvement
  - extremely valuable to government (and thus this company)
- Enables many new uses, deployment on more numerous smaller airborne platforms, even personal systems
- $B scale market

CUSTOMER DISCUSSIONS

2. BIG CONSUMER PRODUCTS COMPANY

- Have internally proven business case for
  at least several vision-based consumer products
- Need small, super-cheap vision system (~100 Gflops)
- One small Singular chip looks adequate
  - cost, in their production volume, is a few dollars
- They say will enable at least several opportunities,
  each with $few-100M annual wholesale revenue
- $B scale market

OTHER PROMISING DOMAINS

(IN PROGRESS - INITIAL EVIDENCE)

- Vision: segmenting smooth objects (weak features, Hartmut/Joe intuition)
- Molecular dynamics, Protein folding (all-atom energy)
- Genomics (eg Smith-Waterman dynamic programming)
- Machine learning (neural nets, genetic algorithms with local crossover, local graphical models, simulated annealing)
- Speech recognition (HMMs, many concurrent voice streams, Dragon CTO)
- Neocortex sim (>human, faster than realtime, supercomputer $)

APPLICATIONS/MARKETS STATUS

- Appear to have at least three $B-scale markets:
  optimization, vision-based cheap consumer products, vision for defense
- At least one "big hammer" technical solution - nearest neighbor
- Likely have web-scale image processing technology
  - feature-based vision (but low power, fast, reliable & cheap/cycle)
  - maybe segmentation (Hartmut says next big thing)
- Specific evidence for several more: speech, learning, bio-tech,
  and crazy stuff - like super-human scale cortex simulation
- General evidence that bright people using the hardware, as I have:
  - will extract additional proven applications from work of the 80s
  - will create unforeseen new opportunities

[redacted]

[redacted]